連載「プロマネの現場から」

第２１４回　システム構築の生産性 2 倍への挑戦

蒼海憲治（大手 SI 企業・グループ会社・事業部長）

私が所属する SI ベンダーの企業グループでは、「システム構築の生産性を 6 年間で 2 倍にする」という目標を掲げ、全社的な変革に向けた取り組みを進めようとしています。ここでいう生産性向上は、単なる工数削減や効率化ではありません。開発の進め方・判断の仕方・組織の動き方そのものを変える、構造的な変革です。

背景には、IT 人材不足、案件の複雑化、顧客ニーズの多様化、技術進化の加速があります。従来の延長線上のやり方では、もはや持続的な競争力を保てない。そうした危機感が、この挑戦の出発点です。

本稿では、現時点での試行や検証の手応えも踏まえながら、「これから重点的に取り組んでいく施策」を、PM の実務視点で整理してみたいと思います。

1. 生成 AI の戦略的活用

―― 魔法ではなく、「使いこなす覚悟」が問われる

生成 AI は、システム構築の生産性を大きく押し上げる可能性を持っています。

一方、PM として現場を見ていると、「使えば自動的に楽になる」ほど単純ではないことも、早い段階で見えてきました。

PoC を進める中で、最初に直面したのは、

「どこまで AI に任せ、どこから人が責任を持つのか」

という線引きの問題です。

たとえばコード生成支援。

CRUD や API 連携といった定型領域では効果が期待できる一方で、

・生成コードのレビュー責任は誰が持つのか

・不具合が起きた場合、顧客にどう説明するのか

こうした点を曖昧にしたままでは、現場は動きません。PM として重要なのは、「効率が上がる」より先に、責任と判断の境界線を明確にすることでした。

ドキュメント生成についても同様です。

「AI が設計書を書く」と表現した瞬間、現場には警戒感が生まれます。

そこで私たちは、「完成品を作らせる」のではなく、「たたき台を作らせ、人が判断する」という位置づけを明確にしました。

テスト観点生成やナレッジ検索でも、生成 AI は判断の代替ではありません。
判断を早める補助輪として位置づけ、その前提を顧客・現場と丁寧に共有していく。
ここに PM の合意形成力が問われます。

2. テスト自動化の徹底
―― 「自動化できる技術」より、「維持できる運用」

テスト自動化は、生産性向上の王道施策です。
しかし PM として何度も見てきたのは、
「最初はうまくいくが、途中で使われなくなる自動化」です。

原因はほぼ共通しています。

・作った人しか直せない
・仕様変更に追従できない
・結局、手動テストが残る

こうした失敗を避けるため、私たちは
「自動化すること」よりも、「誰が・いつ・どう保守するか」を先に決めることを重視しています。

UI テスト自動化では、
「すべて自動化しない」という判断を、PM として意識的にしています。
変更頻度が高い画面を無理に自動化すると、かえって保守工数が増える。
どこを自動化し、どこを割り切るかは、技術判断ではなくプロジェクト判断です。

性能・負荷テストについても、
「ツールはあるが、回していない」状態にならないよう、WBS に組み込み、誰の作業として実行されるのかを明確にします。

3. 標準化と再利用性の追求
―― 「縛る標準」ではなく、「迷わなくて済む標準」

標準化は、現場から最も反発を受けやすい施策です。
PM としてよく聞くのは、
「案件ごとに事情が違う」「標準は現場を分かっていない」
という声です。

だからこそ私たちは、標準化を
「自由を奪うルール」ではなく、「考えなくていい範囲を増やす仕組み」
として位置づけています。

共通フレームワークやテンプレートも、「使え」ではなく、「使うと楽になる」ことを実案件で示すことを重視しています。

マイクロサービスについても、
PM としては慎重にならざるを得ません。
分割すれば速くなるわけではなく、
・チーム分割
・運用体制
・顧客理解
までを含めて、初めて成立します。
そのため、効果が出る条件が揃った案件から試す進め方を取ろうとしています。

ナレッジマネジメントも同様で、
「登録しなさい」と言っても現場は動きません。
「あとから助けられた体験」を作ることが、最大の推進力になります。

4. CI/CD 高度化と DevOps 文化
―― 「パイプライン」より「関係性」

CI/CD や DevOps は技術の話として語られがちですが、
PM の立場から見ると、本質は組織と関係性です。

ツールを入れただけでは、
・リリース判断が遅い
・障害時に責任の押し付け合いが起きる
といった問題は解消しません。

重要なのは、
「誰が、どこまでを、どのタイミングで判断するのか」を明確にすることです。

自動デプロイも、技術的に可能だから進めるのではなく、事故時の説明責任まで含めて整理し、顧客・運用部門と段階的に信頼を積み上げていく必要があります。

DevOps 文化も、形だけでは根付きません。
一緒に障害対応を経験することで、初めて文化になります。

おわりに ―― PM の仕事は「積み上げ続けること」

生産性 2 倍という目標は、新技術を導入すれば達成できるものではありません。

・技術の期待値を適切に下げる
・責任と判断の線を明確にする
・現場・顧客・組織の合意を積み重ねる

その地道な積み重ねこそが、PM に求められる仕事だと感じています。

小さな改善を、現場で「続く形」にする。
その積み重ねが、6 年後の生産性 2 倍につながる――
私はそう信じています。

最後に、生成 AI による「毎月 1%の成長で、生産性が 2 倍になる」ことを意味するキャッチコピーを紹介します。

・小さな一歩、大きな飛躍。月 1%成長で 6 年後の生産性 2 倍を実現。
・複利の力、開発にも。毎月 1%の積み重ねが、6 年で驚異の 2 倍へ。
・焦らず着実に。月 1%の進化が、6 年後の圧倒的な成果を生む。
・魔法ではない、積み重ねの力。月 1%の成長が、生産性 2 倍の現実へ。
・6 年後の未来を変える。今日から始める月 1%の生産性向上。
・見えない成長が、確かな未来を築く。月 1%、6 年後の 200%へ。
・コツコツと、でも確実に。月 1%の改善が、6 年後の革新に繋がる。
・千里の道も一歩から。システム開発の生産性 2 倍も、月 1%から。
・成長のエンジン、月 1%。6 年後のゴールは、生産性 2 倍。
・スモールスタート、ビッグリターン。月 1%の成長投資で、6 年後に 2 倍の成果。

読者の皆様の現場においても、
「小さく始めて、積み上げる」挑戦のヒントになれば幸いです。